


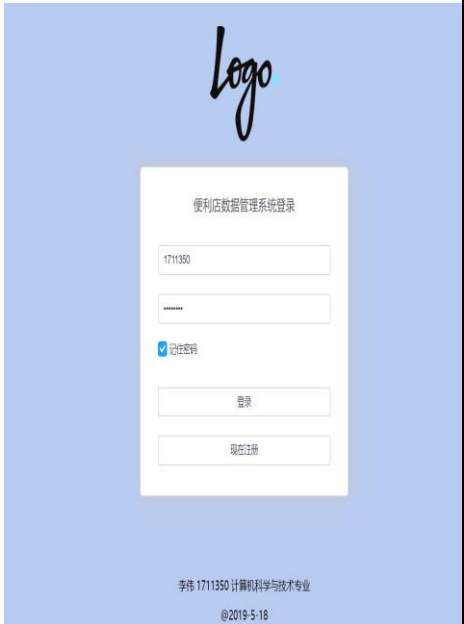
数据库工程作业

要求:

- 1. 完成一个小型的数据库信息管理系统（或部分功能），并填写工程作业报告；程序和报告请在规定时间之内上传。
- 2. 开发模式（B/S 或 C/S）、开发高级语言任选，后台数据库使用大型数据库管理系统（SQL Server、Oracle、MySQL 等），不要使用桌面数据库。
- 3. 报告中所列举的四种操作，每种操作举一个例子即可。
- 4. 作业成绩按照报告中的标准评分，程序只实现报告中涉及的部分即可。
- 5. 作业完成后，请将工程作业报告和程序上传到学院网站，并联系助教老师进行系统说明和演示，回答相关问题。

工程作业报告

1. 项目信息（10 分）

学号	1711350	姓名	李伟	专业	计算机科学与技术专业
项目名称	便利店数据管理系统				
必备环境	Node.js 环境，SQLServer2017				
系统主要功能简介（4 分）	<p>1、基于 SQLServer2017 建立便利店数据库，通过多张关系表保存便利店日常经营数据；</p> <p>2、基于 node.js 以及 Vue 前端框架搭建前端可视化操作平台，实现便利店日常销售数据以及商品库存数据管理，查询，更改，统计等功能，同时能够对商店商品数据进行查询更改，同时实现对于便利店的人员管理以及客户信息维护和管理，以便更好地为核心客户进行服务。</p>				
系统主要页面截图（6 分）	<p>1、注册及登录页面：</p> <div></div> <div></div>				

2、商品销售页面：

Logo

便利店数据库管理系统 登录/注册

销售前台

商品销售

销售业绩

后台管理

库存管理

人员管理

商品名称

商品编号

商品编号

查询

重置

导出数据

<input type="checkbox"/>	商品编号	商品名称	商品标价	生产日期	保质期	剩余数量	促销折扣	限购数量	操作
<input type="checkbox"/>	864010000001	百事可乐	2.5	2019-04-15	2020-08-01	4500	无折扣	不限购	编辑
<input type="checkbox"/>	864010000002	可口可乐	2.5	2019-04-15	2020-08-01	990	无折扣	不限购	编辑
<input type="checkbox"/>	864010000003	康师傅牛肉面	4.5	2019-05-07	2019-12-17	500	无折扣	不限购	编辑
<input type="checkbox"/>	864010000005	特仑苏牛奶	65	2019-05-27	2020-01-21	200	无折扣	不限购	编辑

共 0 条 2 条/页 < > 前往 1 页

3、销售业绩页面：

Logo

便利店数据库管理系统 登录/注册

销售前台

商品销售

销售业绩

后台管理

库存管理

人员管理

时间下限

2019-05-08

时间上限

2019-05-27

查询

业绩统计

导出数据

<input type="checkbox"/>	订单编号	客户编号	商品名称	销售数量	销售价格	进货价格	订单日期	退货日期
<input type="checkbox"/>	1074	020201905208111	可口可乐	5	2.5	1.8	2019-05-25	
<input type="checkbox"/>	1075	020201905208111	可口可乐	5	2.5	1.8	2019-05-25	
<input type="checkbox"/>	1076	020201905208111	可口可乐	5	2.5	1.8	2019-05-25	
<input type="checkbox"/>	1077	020201905208111	可口可乐	5	2.5	1.8	2019-05-25	
<input type="checkbox"/>	1078	020201905208111	可口可乐	5	2.5	1.8	2019-05-25	
<input type="checkbox"/>	1079	020201905208111	可口可乐	5	2.5	1.8	2019-05-25	
<input type="checkbox"/>	1082	020201905208111	可口可乐	5	2.5	1.8	2019-05-27	
<input type="checkbox"/>	1083	020201905208100	百事可乐	500	2.5	1.8	2019-05-27	
<input type="checkbox"/>	1094	020201905208100	特仑苏牛奶	5	65	50	2019-05-28	
<input type="checkbox"/>	1095	020201905208100	康师傅牛肉面	50	4.5	2.3	2019-05-28	
<input type="checkbox"/>	1096	020201905208111	可口可乐	43	2.5	1.8	2019-05-28	

共 500 条 2 条/页 < 1 2 3 4 5 6 ... 250 > 前往 1 页

4、库存管理页面：

Logo

便利店数据库管理系统 登录/注册

销售前台

商品销售

销售业绩

后台管理

库存管理

人员管理

商品名称

商品编号

商品编号

查询

重置

导出数据

库存管理

商品库存信息

商品编号

864010000003

商品名称

康师傅牛肉面

生产厂家

康师傅有限公司

商品标价

4.5

进货价格

2.3

生产日期

2019-05-07

保质期

2019-12-17

添加数量

52

<input type="checkbox"/>	商品编号	商品名称	生产厂家	商品标价	进货价格	生产日期	保质期	操作
<input type="checkbox"/>	864010000001	百事可乐	百事集团	2.5	1.8	2019-04-15	2020	添加
<input type="checkbox"/>	864010000002	可口可乐	娃哈哈集团	2.5	1.8	2019-04-15	2020	添加
<input type="checkbox"/>	864010000004	西瓜	开心农场	2.3	1.5	2019-05-27	2019	添加
<input type="checkbox"/>	864010000003	康师傅牛肉面	康师傅有限公司	4.5	2.3	2019-05-07	2019	添加
<input type="checkbox"/>	864010000005	特仑苏牛奶	开心农场	65	50	2019-05-27	2020	添加

2. 系统配置（10 分）

说明		(2 分) 请说明系统配置情况 (后台数据库, 高级语言); (8 分) 请使用连接串连接高级语言和数据库, 并分析字符串的各个部分。			
配置 步骤 2 分	DBMS	1. 后台数据库为 Microsoft SQL Server Enterprise (64-bit) 版本: 13.0.1601.5			
		2. 数据库管理工具: Microsoft SQL Server Management Studio 2018			
	高级 语言	1. HTML、JavaScript			
		2. Node.js, express, vue.js 框架			
连接串 分析 (6 分)		序号	名称	功能说明	取值
		1	User	用户名字段, 表示数据库用户名, 本实验使用 sa 用户名	Sa
		2	Password	数据库连接认证的用户密码	Lio6218@163.com
		3	server	服务器 IP	localhost
		4	Database	数据库名称	CCstore
		5	Options.encrypt	设置选择, 这个属性确定连接是否被加密	True
		6	pool	连接池配置, pool.max 表示池中最大的连接数, pool.min 表示池中最小的连接数, pool.idleTimeoutMillis 表示关闭连接之前的毫秒数	采用默认值: min: 0, max: 10, idleTimeoutMillis: 3000
连接串代码 (截屏) (2 分)		<pre>const config={ user:'sa', password:'lio6218@163.com', server:'localhost', database:'CCStore', // requestTimeout: 3000, // connectionTimeout:3000, pool: { min: 0, max: 1000, idleTimeoutMillis: 3000 } };</pre>			
备注		本实验服务器后端是基于 express 框架搭建。			

3. 数据库设计（14 分）

说明	（10 分）按照数据表的创建顺序，依次给出所涉及数据表的信息，其中参照字段以“（字段 1，字段 2，……，字段 n）”的形式给出，被参照字段以“表名（字段 1，字段 2，……，字段 n）”的形式给出；
----	--

(4分) 一般 DBMS 都可以为数据库生成关系图, 请将该图片截屏并粘贴到表格中。					
创建顺序	数据表名称	主键	参照属性	被参照表及属性	
1	Commodity	Cmd_ID	Manf_ID	Manufacture (Manf_ID)	
2	DiscountCommodity	Cmd_ID	Cmd_ID	Commodity (Cmd_ID)	
3	Employee	Employee_ID	Store_ID	Store (Store_ID)	
4	FreshCommodity	Cmd_ID	Cmd_ID	Commodity (Cmd_ID)	
5	LimitCommodity	Cmd_ID	Cmd_ID	Commodity (Cmd_ID)	
6	Manufacture	Manf_ID			
7	Register	(VIP_ID, Store_ID)	VIP_ID, Store_ID	Store (Store_ID), VIP (VIP_ID)	
8	Sale	Sale_orderID	VIP_ID, Cmd_ID	Store (Store_ID), Commodity (Cmd_ID)	
9	Store	Store_ID			
10	Supplier	Supplier_ID			
11	Supply	(Store_ID, Supplier_ID, Cmd_ID)	Store_ID, Supplier_ID, Cmd_ID	Store (Store_ID), Commodity (Cmd_ID), Supplier (Supplier_ID)	
12	VIP	VIP_ID			
13	Login	UserID			
14	DeletedSale	(Sale_orderID, Sale_DeleteTime)			
15	DeletedSupply	(Supply_DeleteTime, Store_ID, Cmd_ID, Supplier_ID)			

数据表 (10)

关系图 (4)

备注	DeletedSale 和 DeletedSupply 表是保存删除数据的表格，故没有设置外键
----	---

4. 含有事务应用的删除操作（13 分）

说明	（1 分）简要说明该操作所要完成的功能； （2 分）该操作会涉及的表（必须含有两张或两张以上的关系表，同时以“表名”的形式给出） （1 分）表连接涉及字段描述（描述方式为“表 1. 属性=表 2. 属性”） （1 分）删除条件涉及的字段描述（以“表名. 属性=? ”形式给出） （4 分）实现该操作的关键代码（高级语言、SQL），截图即可；（其中如果删除语句中不包含任何形式的事务应用将扣除 3 分） （4 分）如何执行该操作，按所述方法能够正常演示程序则给分。	
功能描述（1 分）	删除商品中的过期商品或者下架的商品，在删除的同时将依赖于该商品的位于其他数据表中的数据一同删除，部分重要信息进行转存，并利用事务操作保证删除的正确性。	
涉及的表（2 分）	Commodity, Sale, Supply, DiscountCommodity, FreshCommodity, LimitCommodity	
表连接涉及字段（1 分）	Commodity.Cmd_id=Sale.Cmd_id, Commodity.Cmd_id=Supply.Cmd_id, Commodity.Cmd_id=DiscountCommodity.Cmd_id, Commodity.Cmd_id=FreshCommodity.Cmd_id, Commodity.Cmd_id=LimitCommodity.Cmd_ID	
删除条件字段描述（1 分）	字段	规则
	Commodity.Cmd_id=?	Cmd_id 为小于等于 40 位的字符串数据
	...	
代码（4 分）	（截屏） <pre> begin tran declare @error int,@cmd_id varchar(40); set @error=0; set @cmd_id='864010000004'; begin try --删除前先将Sale中以及Supply表中的数据转存 insert into DeletedSale select Sale_orderID, getdate(), VIP_ID ,Cmd_ID , Sale_time ,Sale_num ,Sale_price ,Sale_returnTime ,Sale_returnPrice from sale where Cmd_ID=@cmd_id; insert into DeletedSupply select getdate(),Store_ID,Cmd_ID ,Supplier_ID ,Supply_time,Supply_num from supply where Cmd_id=@cmd_id; --删除子类中数据 delete from sale where Cmd_id=@cmd_id; delete from Supply where Cmd_ID=@cmd_id; delete from DiscountCommodity where DiscountCommodity.Cmd_ID=@cmd_id; delete from LimitCommodity where LimitCommodity.Cmd_ID=@cmd_id; delete from freshCommodity where FreshCommodity.Cmd_ID=@cmd_id; delete from commodity where cmd_id=@cmd_id; end try begin catch set @error=@error+1; end catch end catch if (@error!=0) begin rollback tran print @error end else begin commit tran end </pre>	
程序演示（4 分）	<ul style="list-style-type: none"> 在库存商品页查询出来的数据项后面的操作按钮中点击删除按钮能够实现对商品的删除。 删除前数据页： 	

商品名称

商品名称

商品编号

商品编号

查询

重置

导出数据

库存管理

导出数据

添加库存

商品编号

864010000004

商品名称

特仑苏

生产厂家

特仑苏有限公司

商品标价

65

进货价格

45

生产日期

2019-05-01

保质期

2019-12-17

添加数量

500

商品库存信息

<input type="checkbox"/>	商品编号	生产厂家	商品标价	进货价格	生产日期	保质期	操作
<input type="checkbox"/>	864010000002	可口可乐有限公司	2.5	1.8	2019-05-01	2019-12-17	添加 删除
<input type="checkbox"/>	864010000004	特仑苏有限公司	65	45	2019-05-01	2019-12-17	添加 删除
<input type="checkbox"/>	864010000003	康师傅	4.5	2.3	2019-05-07	2019-12-17	添加 删除
<input type="checkbox"/>	864010000001	康师傅有限公司	4.5	2.3	2019-05-07	2019-12-17	添加 删除

共 500 条2 条/页123456...250>第 1 页

● 删除后的库存管理数据：

✓

操作成功!

便利店数据库管理系统

登陆/注册

商品编号

查询

重置

导出数据

商品库存信息

<input type="checkbox"/>	商品编号	商品名称	生产厂家	商品标价	进货价格	生产日期	保质期	操作
<input type="checkbox"/>	864010000002	可口可乐	可口可乐有限公司	2.5	1.8	2019-05-01	2019	添加 删除
<input type="checkbox"/>	864010000003	康师傅牛肉面	康师傅有限公司	4.5	2.3	2019-05-07	2019	添加 删除
<input type="checkbox"/>	864010000001	苹果	康师傅有限公司	4.5	2.3	2019-05-07	2019	添加 删除

备注

5. 触发器控制下的添加操作（20 分）

说明	（1 分）简要说明该操作所要完成的功能； （2 分）简要说明该触发器所要完成的功能 （1 分）该操作会涉及的表（以“表名”的形式给出）。 （2 分）该操作输入数据以及输入数据应该满足的条件，如：数值范围、是否为空； （6 分）实现该操作的关键代码（高级语言、SQL），截图即可； （8 分）如何执行该操作，按所述方法能够正常演示程序则给分。	
功能描述 （1 分）	在销售页面，可以对商品进行任何数量的销售，并将销售数据记录到 Sale 表中，同时能够自动更新客户信息以及商品的库存数量。	
触发器描述 （2 分）	在插入 Sale 表中销售订单记录之前，对 Commodity 和 VIP 表中的数据进行更新，更新成功之后再销售记录插入到 Sale 数据表中	
涉及的表 （1 分）	Sale, VIP, Commodity	
输入数据 （2 分）	字段	规则
	VIP_ID	varchar(40), not null
	Cmd_ID	varchar(40), not null
	Sale_orderID	varchar(40), not null
	Sale_price	Float, null
	Sale_returnTime	Datetime, null
	Sale_time	Datetime, null
	Sale_num	Int, null
插入操作 源码 （3 分）	（截屏）  <pre> router.post('/insertSale', function(req, res, next) { console.log(req.body); // Query var sqlquery="insert into sale values('"+req.body.vip_id+"',\n"+ "'"+req.body.cmd_id+"','"+(select max(sale_orderid)+1 from sale)\n"+ "','"+getdate(),""+req.body.sale_num+"','"+req.body.sale_price+"',null,null)"; console.log(sqlquery); db.sql(sqlquery, callback: function (err, result) { if (err) { console.log(res.data); return err; } console.log(result.recordset); res.json(result); }) }); </pre>	

触发器源代码
(3分)

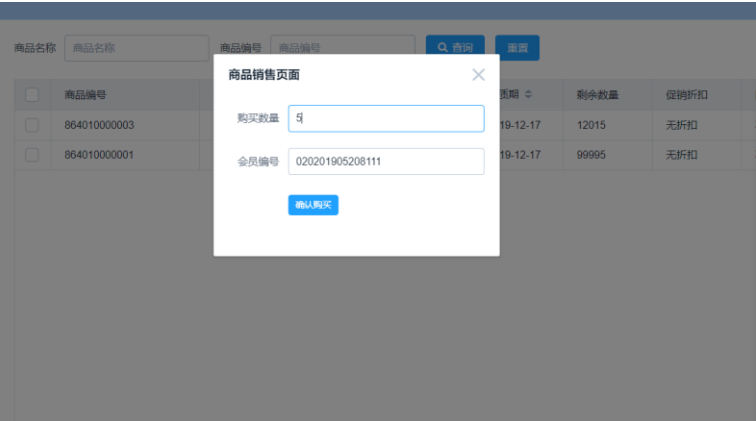
(截屏)

```
alter trigger tgInsertSaleUpdateCommodityAndVIP
on sale INSTEAD OF insert
as
begin tran
    declare @cmd_id varchar(40), @Vip_id varchar(40), @cmd_salenum int, @totalmoney float;
    declare @tran_error int;
    set @tran_error=0;
begin try
    set @cmd_id = (select Cmd_ID from inserted);           --获取商品编号
    set @cmd_salenum=(select Sale_num from inserted);      --获取销售的数量
    set @Vip_id=(select VIP_ID from inserted);            --获取会员ID
    set @totalmoney=(select Sale_price*Sale_num as money from inserted); --获取会员消费金额，等于小商品的购买价格乘以购买数量
end try
begin catch
    set @tran_error=@tran_error+1;
end catch
begin try
    --更新商品的剩余数量
    update Commodity
    set Cmd_leftNum=Cmd_leftNum-@cmd_salenum
    where cmd_ID=@cmd_id and cmd_leftnum=@cmd_salenum=0;    --库存不足的时候不能更新
end try
begin catch
    set @tran_error=@tran_error+10;
end catch
begin try
    --更新VIP客户的消费金额
    update VIP
    set VIP_consumption = VIP_consumption+@totalmoney
    where VIP_ID=@Vip_id;
end try
begin catch
    set @tran_error=@tran_error+11;
end catch
begin try
    insert into sale select * from inserted;    --上述事务进行成功后对sale表进行插入
end try
begin catch
    set @tran_error=@tran_error+12;
end catch
if (@tran_error=0)
begin
    rollback tran;    --回滚事务
    print @tran_error;
end
else
begin
    commit tran;    --回滚事务
    print @tran_error;
end
end
```

程序演示
(4分)

说明：不违背触发器能够执行插入操作。

- 点击销售按钮，输入会员编号和购买数量能够进行销售记录的插入



- 插入成功，能够自动计算价格和总的钱数，并进行提示：



程序演示 (4分)	<p>说明：违背触发器要求，不能够执行插入操作，系统报错。</p> <ul style="list-style-type: none">● 如果输入的销售数量大于库存商品的剩余数，则插入失败，系统进行报错提示  <ul style="list-style-type: none">● 报错页面： 
备注	

6. 存储过程控制下的更新操作（18分）

说明	<p>(1分) 简要说明该操作所要完成的功能；</p> <p>(1分) 简要说明该存储过程所要完成的功能；</p> <p>(2分) 说明该操作涉及操作的表（必须包含两张或两张以上的关系表，以“表名形式”描述）</p> <p>(1分) 表连接涉及字段描述（描述方式为“表 1. 属性=表 2. 属性”）</p> <p>(2分) 该操作会修改字段（以“表名. 字段名”的形式给出），以及修改规则，如新数值的计算方法、在何种条件下予以修改等；</p> <p>(6分) 实现该操作的关键代码（高级语言、SQL），截图即可；</p> <p>(5分) 如何执行该操作，按所述方法能够正常演示程序则给分。</p>
功能描述 (1分)	更新商品库存，实现进货和管理库存的功能，在更新库存的同时能够判断添加商品是否已经存在，否则将作为新的数据元组插入表中，同时对于商品表中的生产厂家进行判断，判断是否厂家是存在于数据库中的否则将对生产厂家的关系表 manufacture 表进行数据插入，如果已经存在于表中则直接对数据项进行更新。
存储过程功能描述	首先判断商品是否是存在于 commodity 数据表中，是则更新数据表，否，则对数据表进行插入操作，同时对关联数据表进行处理。

(1分)		
涉及的关系表(2分)	Commodity, Manufacture	
表连接涉及字段(1)	Commodity.manf_ID=Manufacture.manf_ID	
更改字段(2分)	字段	规则
	Commodity.cmd_Leftnum	当存储过程参数满足例如@cmd_id=Commodity.cmd_id时确定更新的元组项完全符合才对该字段进行修改, 修改规则为该字段数值加上参数@addnum, 即 cmd_leftnum=cmd_leftnum+@addnum

更新代码(3分)	(截屏) <pre> if(@cmd_id in (select Cmd_ID from Commodity left join Manufacture on Commodity.Manf_ID=Manufacture.Manf_ID where Cmd_ID=@cmd_id and Manf_name=@cmd_manf and Cmd_name= @cmd_name and Cmd_saleprice=@cmd_saleprice and Cmd_buyprice=@cmd_buyprice and Cmd_proDate=@cmd_prodate and Cmd_shelflife=@cmd_shelflife)) begin begin try update commodity set Cmd_leftNum=Cmd_leftNum+@addnum where Cmd_ID=@cmd_id; end try begin catch set @error=@error+1; end catch if(@error>0) begin rollback tran end else begin commit tran end end </pre>	

创建
存储
过程
源码
(3
分)

(截屏)


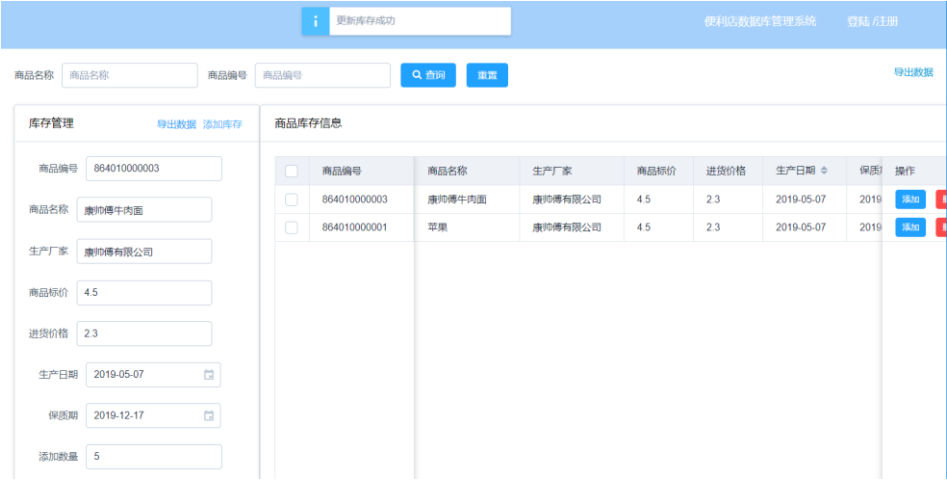
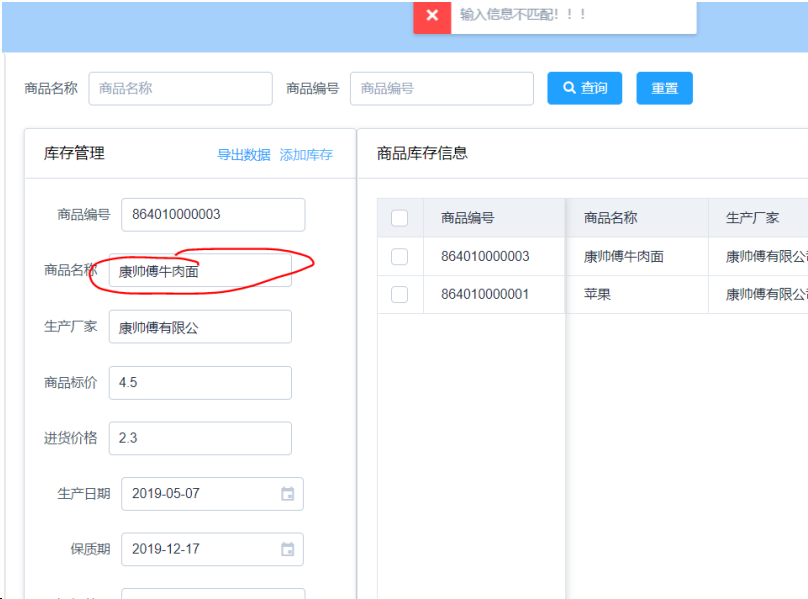
```
if(exists (select * from sys.objects where name='spUpdateCommodity'))
    drop proc spUpdateCommodity;
go
create proc spUpdateCommodity(@cmd_id varchar(40),@cmd_name varchar(50),@cmd_manf varchar(50),
@cmd_saleprice float,@cmd_buyprice float,@cmd_prodate datetime ,@cmd_shelflife datetime,@addnum int)
as
begin tran
    declare @error int;
    set @error=0;
    if(@cmd_id not in (select cmd_id from Commodity) )
    begin
        begin try
            if(@cmd_manf not in(select Manufacture.Manf_name from Manufacture))
                insert into Manufacture values( ('#' +cast(datediff(ss,'12/25/2006',getdate()) as varchar(20))),null,null,@cmd_manf)
            insert into commodity values(@cmd_id,(select Manufacture.Manf_ID from manufacture where Manf_name=@cmd_manf)
,@cmd_name,@cmd_saleprice,@cmd_buyprice,@cmd_prodate,@addnum,@cmd_shelflife)
        end try
        begin catch
            set @error=@error+1;
        end catch
        if(@error>0)
            begin rollback tran end
        else
            begin commit tran end
    end
else
    begin
        if(@cmd_id in (select Cmd_ID
from Commodity left join Manufacture on Commodity.Manf_ID=Manufacture.Manf_ID
where Cmd_ID=@cmd_id and Manf_name=@cmd_manf and Cmd_name= @cmd_name and Cmd_saleprice=@cmd_saleprice
and Cmd_buyprice=@cmd_buyprice and Cmd_proDate=@cmd_prodate and Cmd_shelflife=@cmd_shelflife))
        begin
            begin try
                update commodity
                set Cmd_leftNum=Cmd_leftNum+@addnum
                where Cmd_ID=@cmd_id;
            end try
            begin catch
                set @error=@error+1;
            end catch
            if(@error>0)
                begin rollback tran end
            else
                begin commit tran end
        end
        else
            begin
                begin try
                    throw 60000, 'In try block.', 1;
                end try
                begin catch
                    set @error=@error+1;
                end catch
                if(@error>0)
                begin
                    print @error
                    rollback tran
                    RAISERROR('输入信息不匹配!!!',16,1)
                end
                else
                begin
                    print @error
                    commit tran
                end
            end
        end
    end
end
```

存储
过程
执行
源码
(1
分)

(截屏)

```
router.post('/UpdateCommodity', function( req, res, next) {
    console.log(req.body);
    // Query
    var adder=req.body.adder;
    var sqlquery="exec spUpdateCommodity '"+ adder.cmd_id
    +"', '"+adder.cmd_name+"', '"+adder.cmd_manf+"', '"+
    adder.cmd_saleprice+"', '"+adder.cmd_buyprice+"', '"+adder.cmd_proDate+
    "' , '"+adder.cmd_shelflife+"', '"+adder.addnum;
    console.log(sqlquery);

    db.sql(sqlquery, callBack: function (err, result) {
        if (err) {
            console.log(err);
            return err;
        }
        console.log(result.recordset);
        res.json(err);
    });
});
```

<div>程序演示 (2分)</div>	<div><p>说明：不违背存储过程，能够执行更新操作</p><ul style="list-style-type: none">● 点击添加按钮，实现对库存数据的更新：</div> <div></div> <div><ul style="list-style-type: none">● 成功页面</div> <div></div>
<div>程序演示 (2分)</div>	<div><p>说明：违背存储过程，系统报错：</p><ul style="list-style-type: none">● 当数据不匹配的时候系统进行报错，报错页面如下：</div> <div></div>
<div>备注</div>	

7. 含有视图的查询操作（15 分）

说明	<p>（1 分）简要说明该操作所要完成的功能；</p> <p>（1 分）简要说明建立的该视图的功能；</p> <p>（2 分）简要说明该操作涉及的关系数据表（以“表名”的形式给出）</p> <p>（1 分）简要说明表连接涉及的字段（以“表 1. 属性=表 2. 属性”）</p> <p>（6 分）实现该操作的关键代码（高级语言、SQL），截图即可；</p> <p>（4 分）如何执行该操作，按所述方法能够正常演示程序则给分。</p>
操作功能描述 （1 分）	对商品库存信息进行查询，管理查看商品的进货来源和销售信息，以及促销和限制购买等信息。
视图功能描述 （1 分）	从表 Commodity, DiscountCommodity, FreshCommodity, LimitCommodity, Manufacture 等中查询相关信息组成视图，实现对这些数据的快速查询。
涉及的关系表 （2 分）	Commodity, DiscountCommodity, FreshCommodity, LimitCommodity, Manufacture
表连接字段 （1 分）	<p>Commodity.cmd_id=DiscountCommodity.cmd_ID,</p> <p>Commodity.cmd_id=FreshCommodity.cmd_ID,</p> <p>Commodity.cmd_id=LimitCommodity.cmd_ID,</p> <p>Commodity.cmd_id=Manufacture.cmd_ID</p>

<p>创建视图代码 (3分)</p>	<p>(截屏)</p> <pre> go create view ViewCommodityInfoForManagement as select commodity.cmd_id ,cmd_name,Manufacture.Manf_name as cmd_manf, cmd_saleprice,cmd_buyprice,cmd_proDate,cmd_shelflife,FreshCommodity.FCmd_temp as cmd_temprature, cmd_leftnum,DisCmd_disc as cmd_discount,DisCmd_discStart as cmd_disc_start,DisCmd_DiscEnd as cmd_disc_end, limitCmd_num as cmd_limitnum,LimitCmd_start as cmd_limit_start,LimitCmd_end as cmd_limit_end from (((commodity left join Manufacture on Commodity.Manf_ID=Manufacture.Manf_ID) left join DiscountCommodity on commodity.Cmd_ID=DiscountCommodity.Cmd_ID) left join LimitCommodity on commodity.Cmd_ID=LimitCommodity.Cmd_ID) left join FreshCommodity on Commodity.Cmd_ID=FreshCommodity.Cmd_ID go </pre>
<p>查询代码 (3分)</p>	<p>(截屏)</p> <p>1、查询语句</p> <pre> select * from ViewCommodityInfoForManagement </pre> <p>2、查询操作源码:</p> <pre> router.post('/getInfoForManagement', function(req, res, next) { console.log(req.body); // Query var name=req.body.name; var id=req.body.id; var sql1='select * from ViewCommodityInfoForManagement'; var sql2='select * from ViewCommodityInfoForManagement where cmd_name= '"+name+"'"; var sql3='select * from ViewCommodityInfoForManagement where cmd_name= '"+name+"' and cmd_ID= '"+id+"'"; var sql4='select * from ViewCommodityInfoForManagement where cmd_ID= '"+id+"''; var sqlquery; if(name===' ' && id===' ') sqlquery=sql1; else if(name===' ' && id!==' ') sqlquery=sql4; else if(name!==' ' && id===' ') sqlquery=sql2; else sqlquery=sql3; console.log(sqlquery); db.sql(sqlquery, callBack: function (err, recordset) { if (err) { console.log(err); return; } console.log(recordset.recordset); res.json(recordset.recordset); }); }); </pre>
<p>程序演示 (4分)</p>	<ul style="list-style-type: none"> ● 点击库存管理的上端查询或者重置按钮，能够实现对库存商品信息的查询操作：

商品名称

商品名称

商品编号

商品编号

查询

重置

导出数据

库存管理

导出数据 添加库存

商品编号

864010000004

商品名称

特仑苏

生产厂家

特仑苏有限公司

商品标价

65

进货价格

45

生产日期

2019-05-01

保质期

2019-12-17

添加数量

500

商品库存信息

<input type="checkbox"/>	商品编号	生产厂家	商品标价	进货价格	生产日期	保质期	操作
<input type="checkbox"/>	864010000002	可口可乐有限公司	2.5	1.8	2019-05-01	2019-12-17	<div>添加删除</div>
<input type="checkbox"/>	864010000004	特仑苏有限公司	65	45	2019-05-01	2019-12-17	<div>添加删除</div>
<input type="checkbox"/>	864010000003	两面	4.5	2.3	2019-05-07	2019-12-17	<div>添加删除</div>
<input type="checkbox"/>	864010000001	康帅傅有限公司	4.5	2.3	2019-05-07	2019-12-17	<div>添加删除</div>

共 500 条

2 条/页

<

1

2

3

4

5

6

...

250

>

前往

1

页

备注